RSAC 2025 Conference

Many Voices.
One Community.

SESSION ID: USENIX-M03B

WhisperFuzz: White-Box Fuzzing for Detecting and Locating Timing Vulnerabilities in Processors

Pallavi Borkar

PhD Student
Indian Institute of Technology, Madras
https://www.linkedin.com/in/pallavi-borkar

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

The Big Picture

Hardware Vulnerabilities

- Unprivileged Data Access
- Computing System Compromise

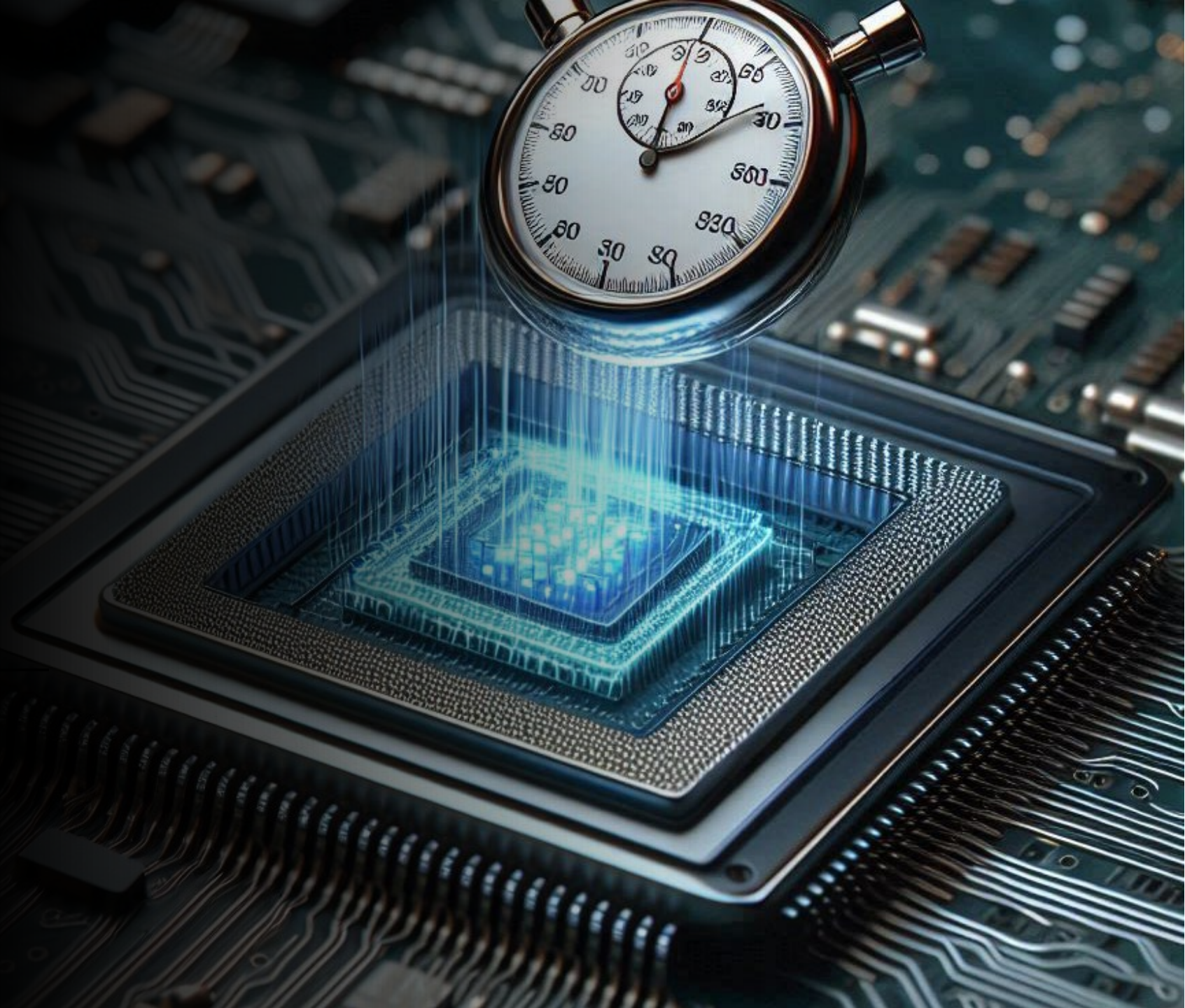
Limitations of Existing Solutions

- No root cause localization
- No effective coverage

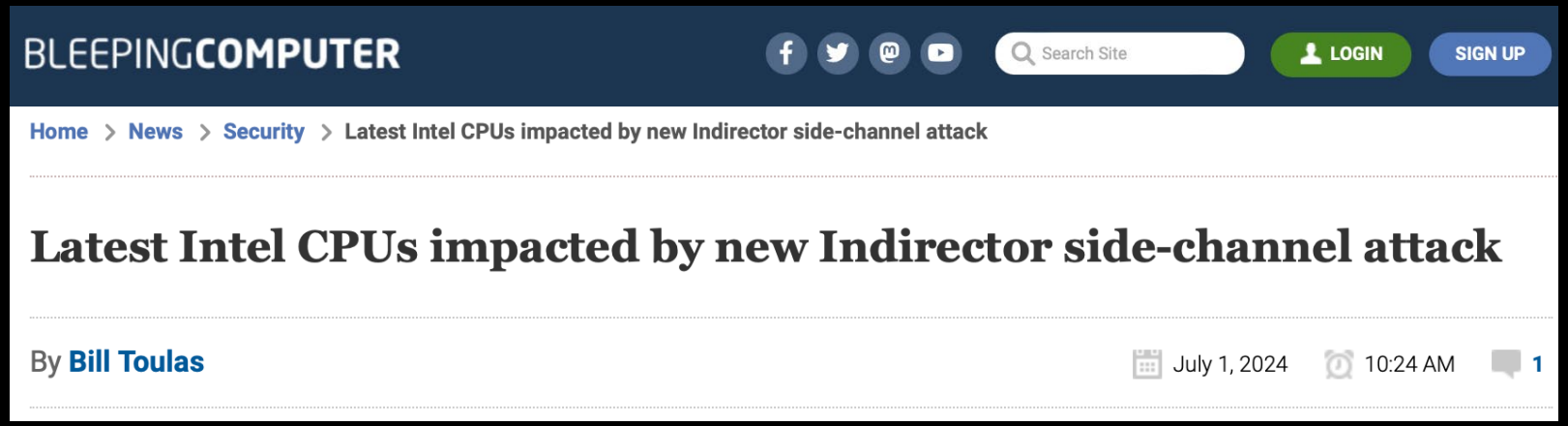
Our Solution, WhisperFuzz

- Automation
- Localizing Leakage
- Effective Coverage

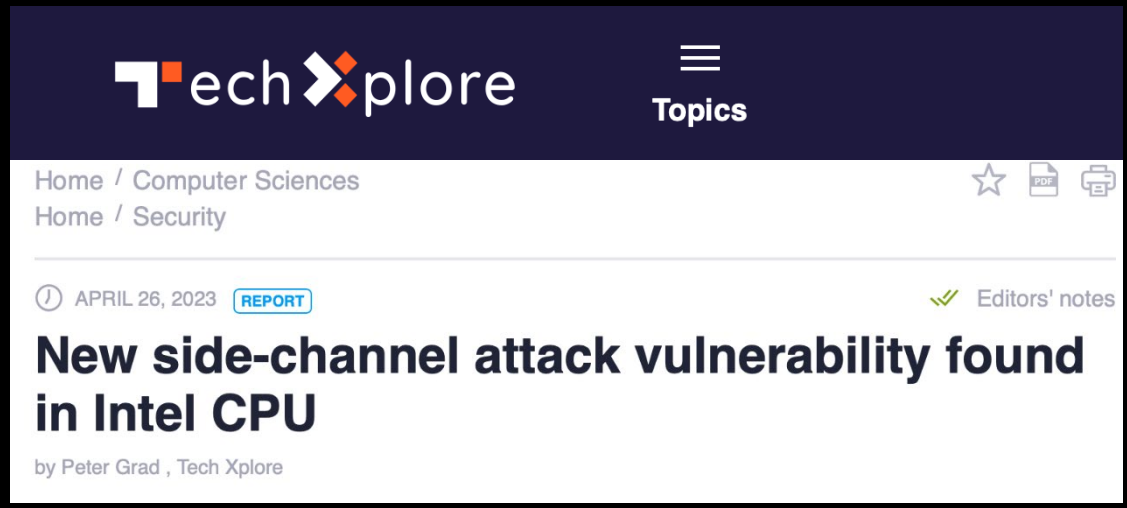
Timing Vulnerabilities



Timing Vulnerabilities: A Crucial Threat



SECURITYWEEK
CYBERSECURITY NEWS, INSIGHTS & ANALYSIS
Malware & Threats
Security Operations
Security Architecture
Risk Management
CISO Strategy
ICS/OT
Funding/M&A
VULNERABILITIES
Researchers Disclose New Side-Channel Attacks Affecting All AMD CPUs
Researchers have disclosed the details of new timing and power-based side-channel attacks that affect all CPUs made by AMD, but the chipmaker says no new mitigations are necessary.
By Eduard Kovacs
October 15, 2021



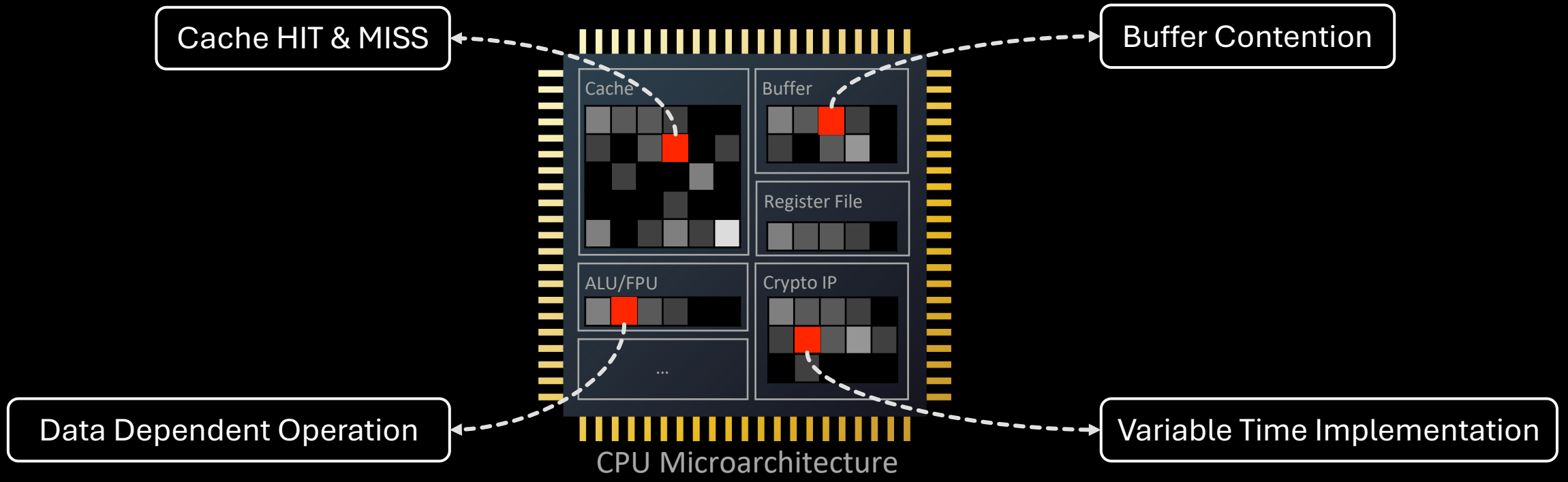
What Are Timing Vulnerabilities

Exponent =
01100010

square_multiply

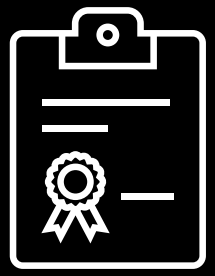
```
def square_multiply(exponent, base):
    output value = 1;
    for each_bit in exponent:
        //Square Step
        value = value * value
        if each_bit == 1:
            //Multiply
            value = value * base
```

Timing Channels: Possible sources



Timing Vulnerability Detection Approaches

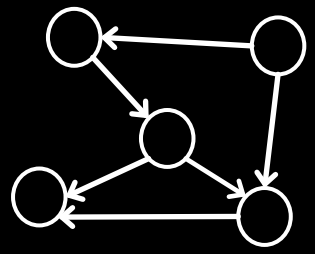
Formal Verification



- State Explosion
- Expert Knowledge

Fadiheh et al., DAC'20

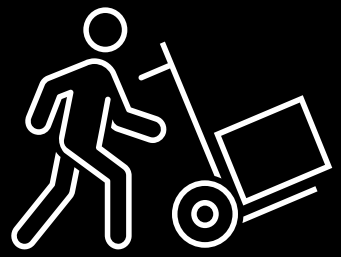
Static Analysis



- Expert Knowledge
- Manual Effort

Trippel et al., Micro '18

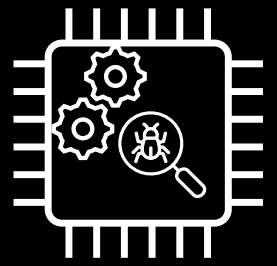
Template Based



- Known Attacks
- Manual Effort

Ibrahim et al., SIGSAC'22

Run-time Detection

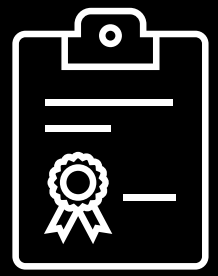


- Post Silicon
- Design Overhead

Gras et al., NDSS '20

Timing Side Channel Detection

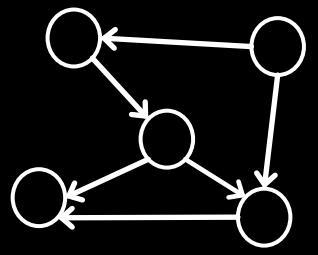
Formal Verification



- State Explosion
- Expert Knowledge

Fadiheh et al., DAC'20

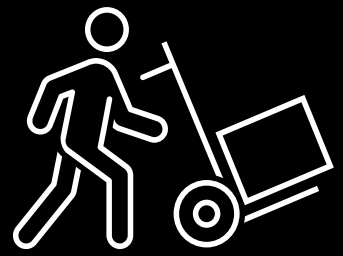
Static Analysis



- Expert Knowledge
- Manual Effort

Trippel et al., Micro '18

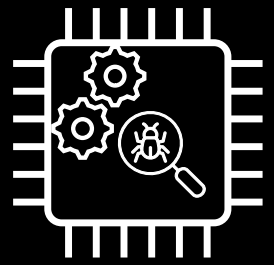
Template Based



- Known Attacks
- Manual Effort

Ibrahim et al., SIGSAC'22

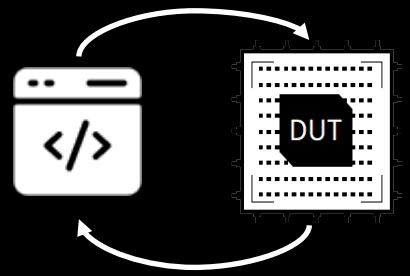
Run-time Detection



- Post Silicon
- Design Overhead

Gras et al., NDSS '20

Hardware Fuzzing



- Scalable
- Automated

Hardware Fuzzing for Timing Vulnerabilities

- Manual effort
- No root cause localization
- No coverage

	FEATURES				
HARDWARE FUZZER	Cannot localize	Manual effort		Limited to cache timing channels	Cannot localize
	Limited sequences	Limited to SE	Cannot localize	Post-silicon	Register coverage metric
	Reverted	Meltdown	Achieved high design coverage	Can detect Spectre style	Automated
	attacks	Pre-silicon	Pre-silicon	Automated	Can detect Spectre style
	Weber et al. USENIX '21	M. Ghaniyoun et al. ISCA '21	R. Kande et al., USENIX SEC '22	O. Oleksenko et al. IEEE S&P '23	Rajapaksha et al. DATE '23

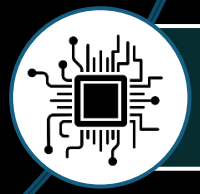
Challenges & Contributions

First automated whitebox fuzzer

Pinpoint root-causes of the timing vulnerability

Novel coverage metric

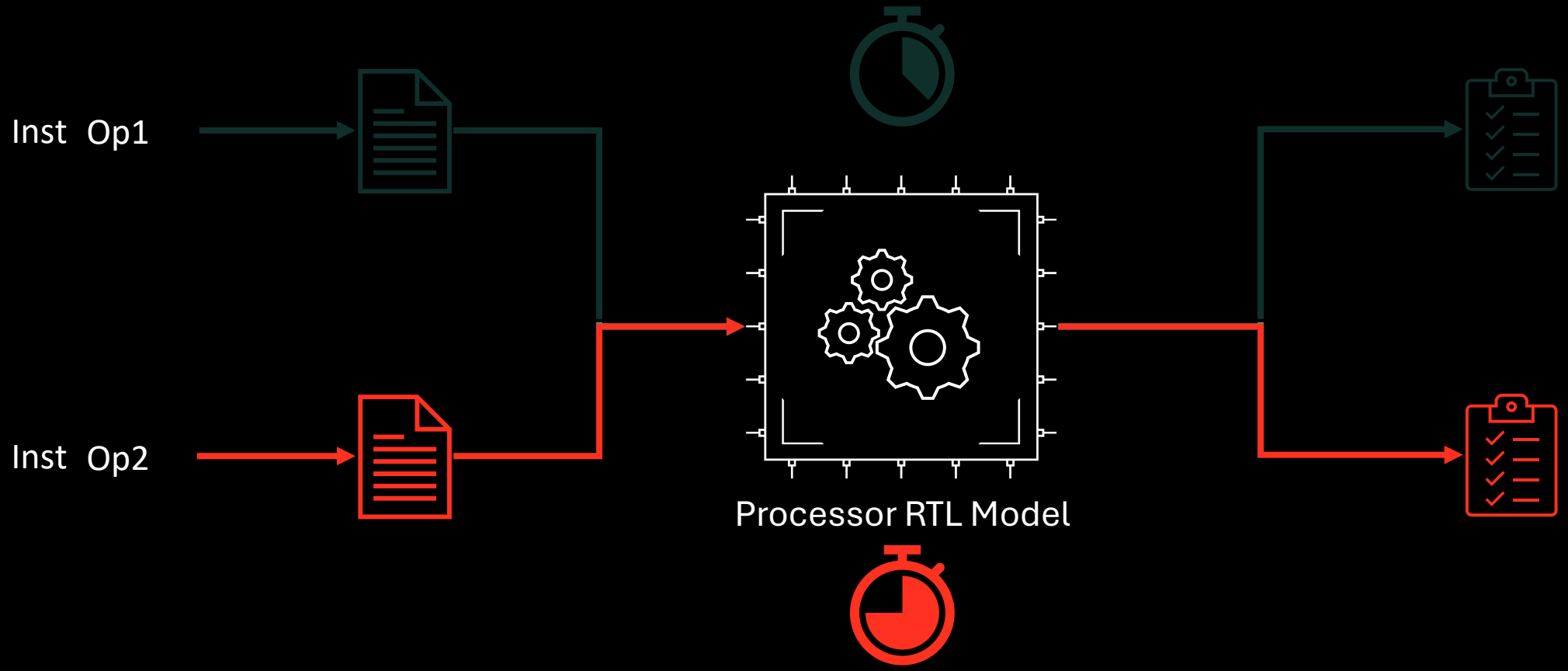
Novel vulnerability applicability



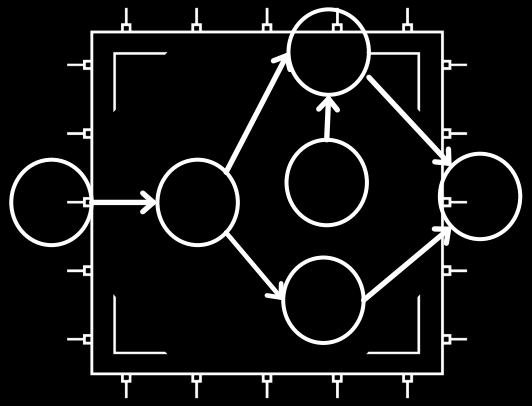
Evaluated on 3 pipelined processors

Our Solution: WhisperFuzz

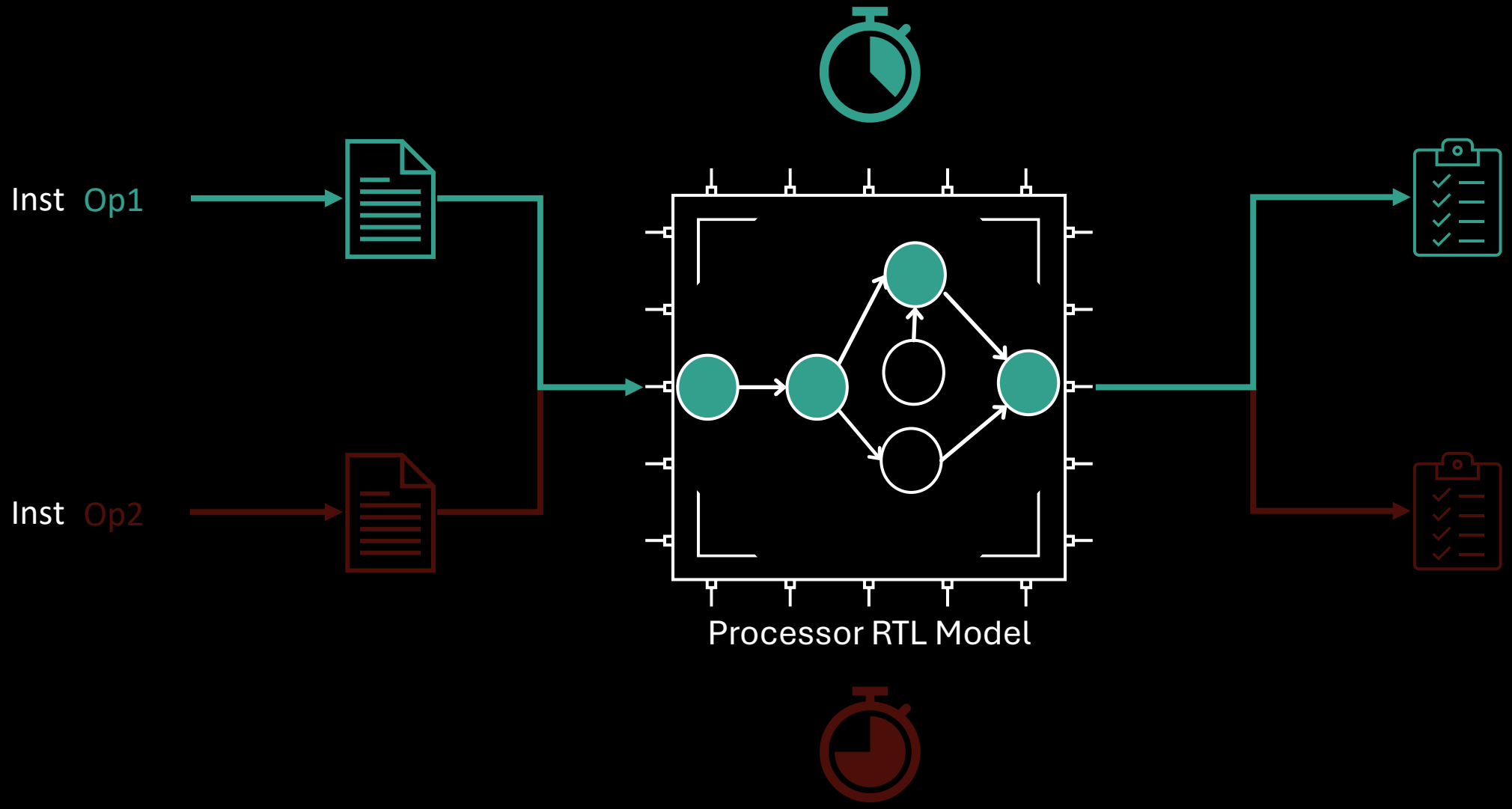
WhisperFuzz: High-Level Idea



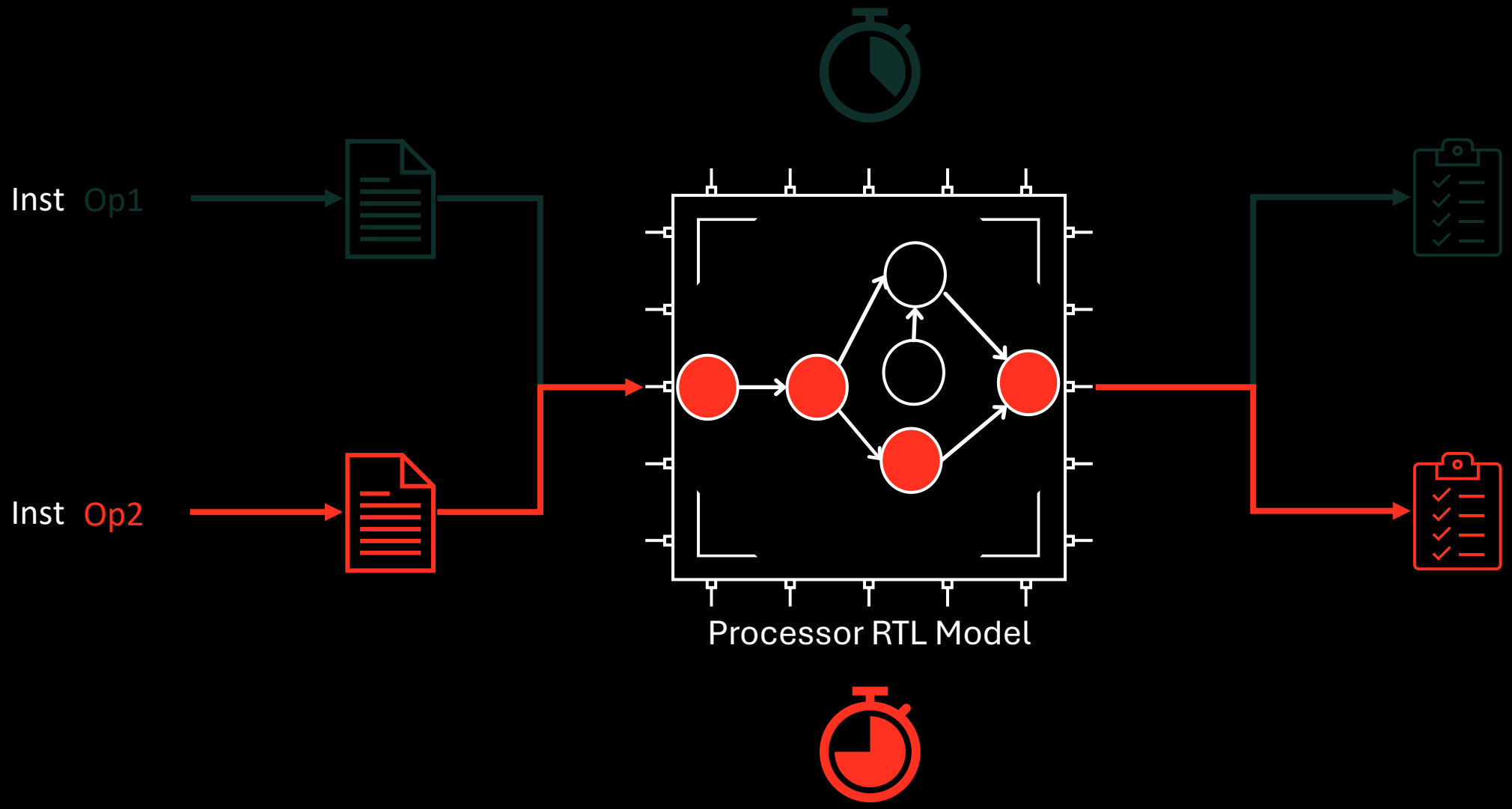
WhisperFuzz: High-Level Idea



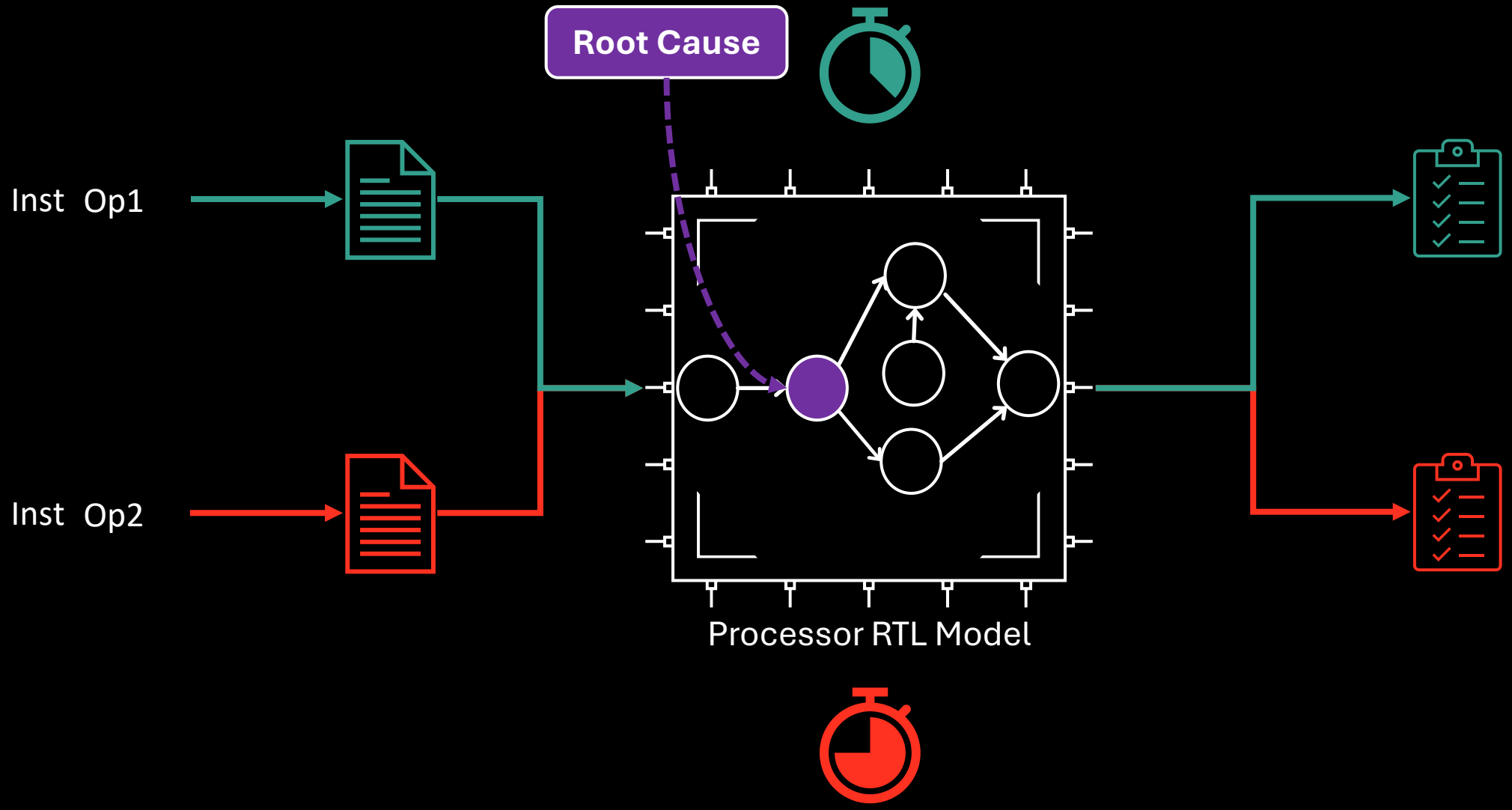
WhisperFuzz: High-Level Idea



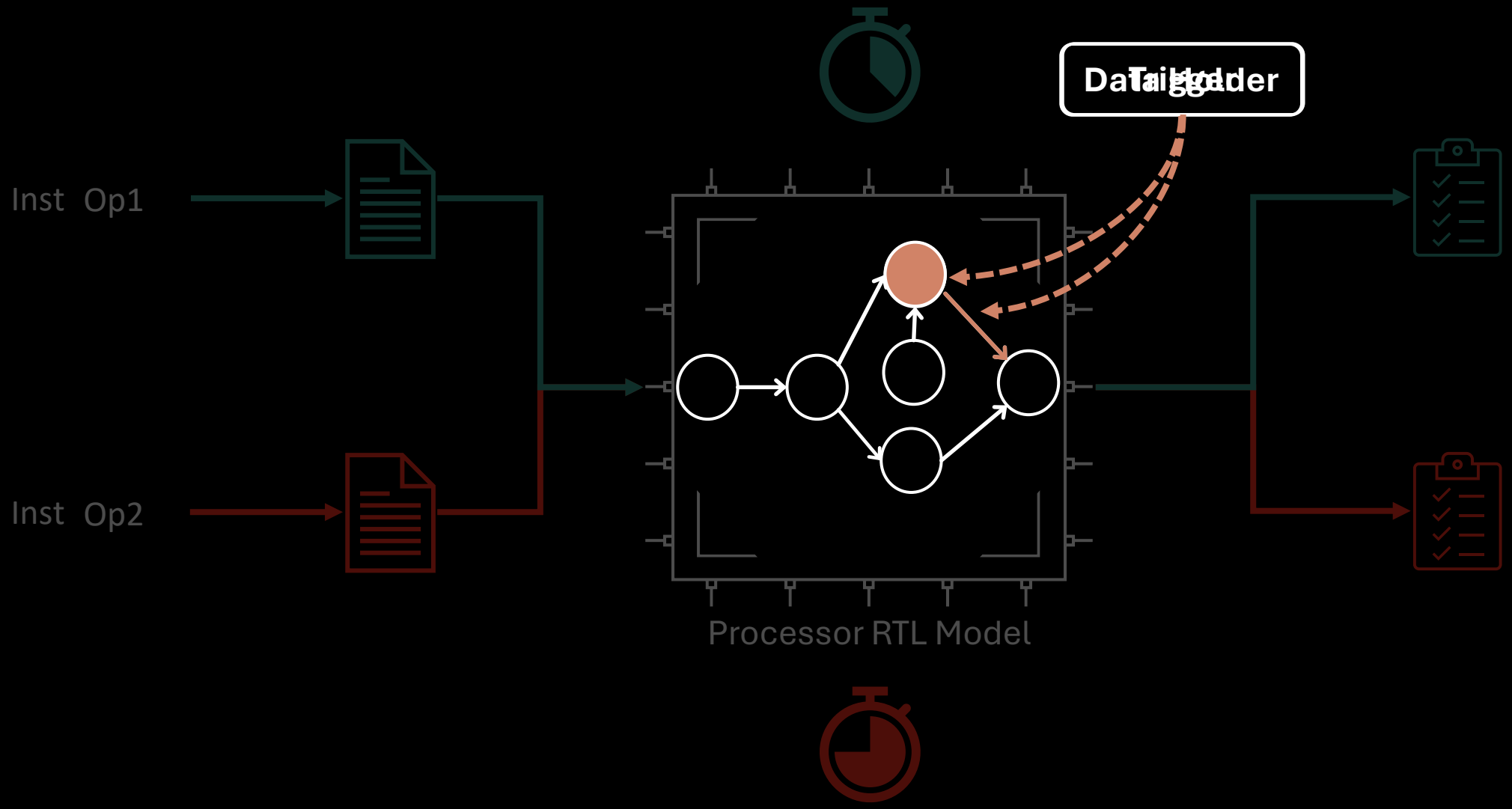
WhisperFuzz: High-Level Idea



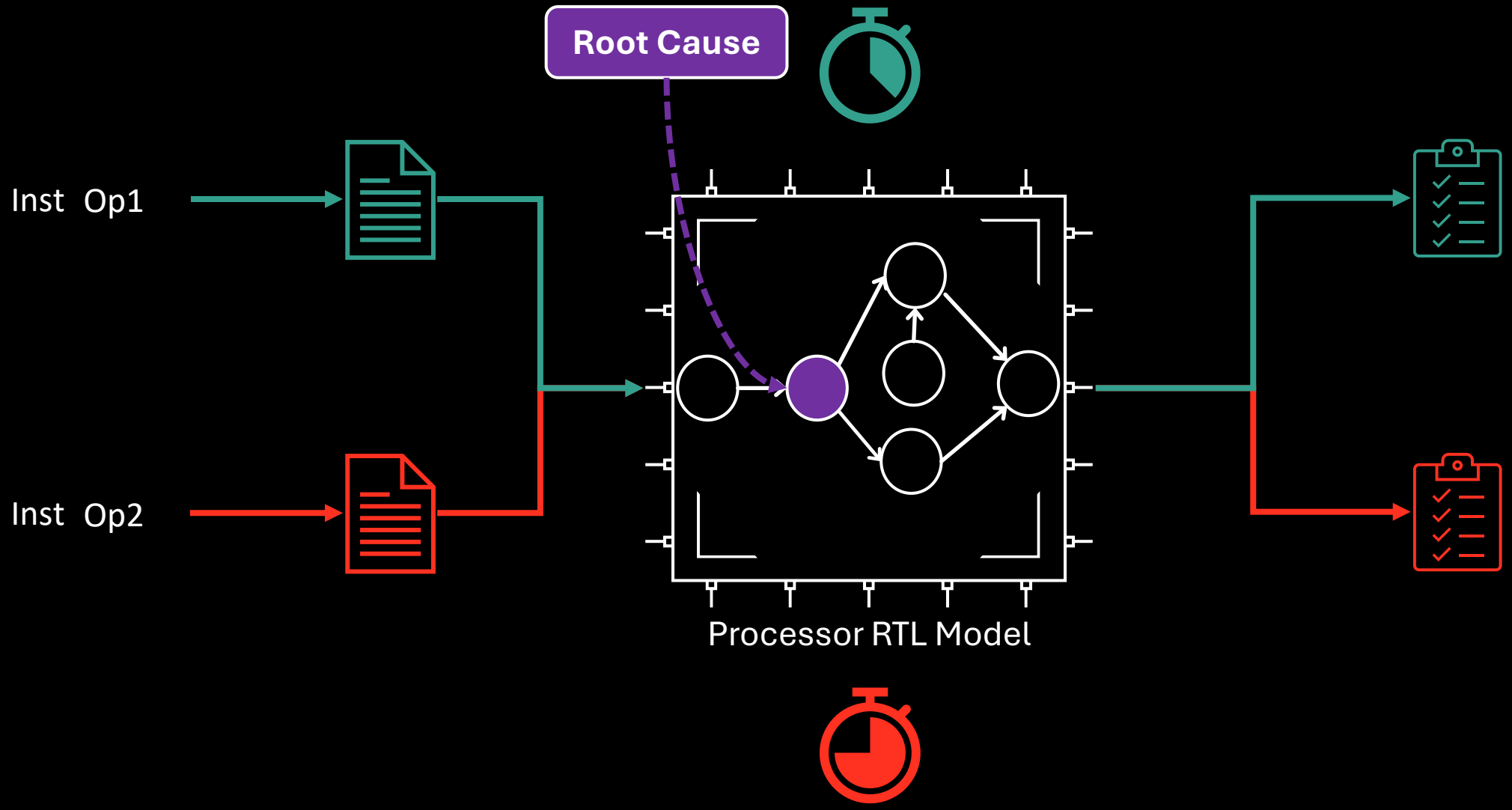
WhisperFuzz: High-Level Idea



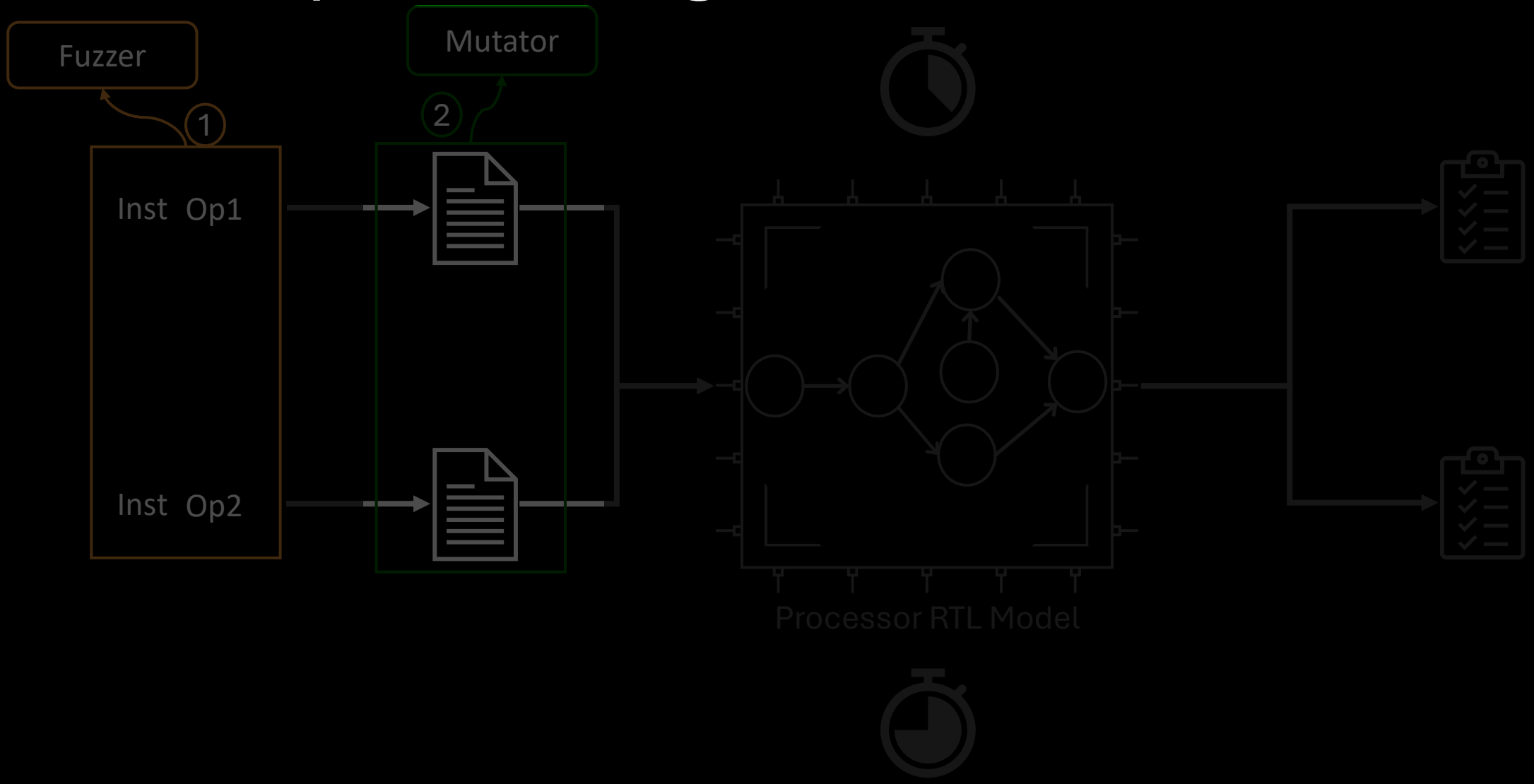
WhisperFuzz: High-Level Idea



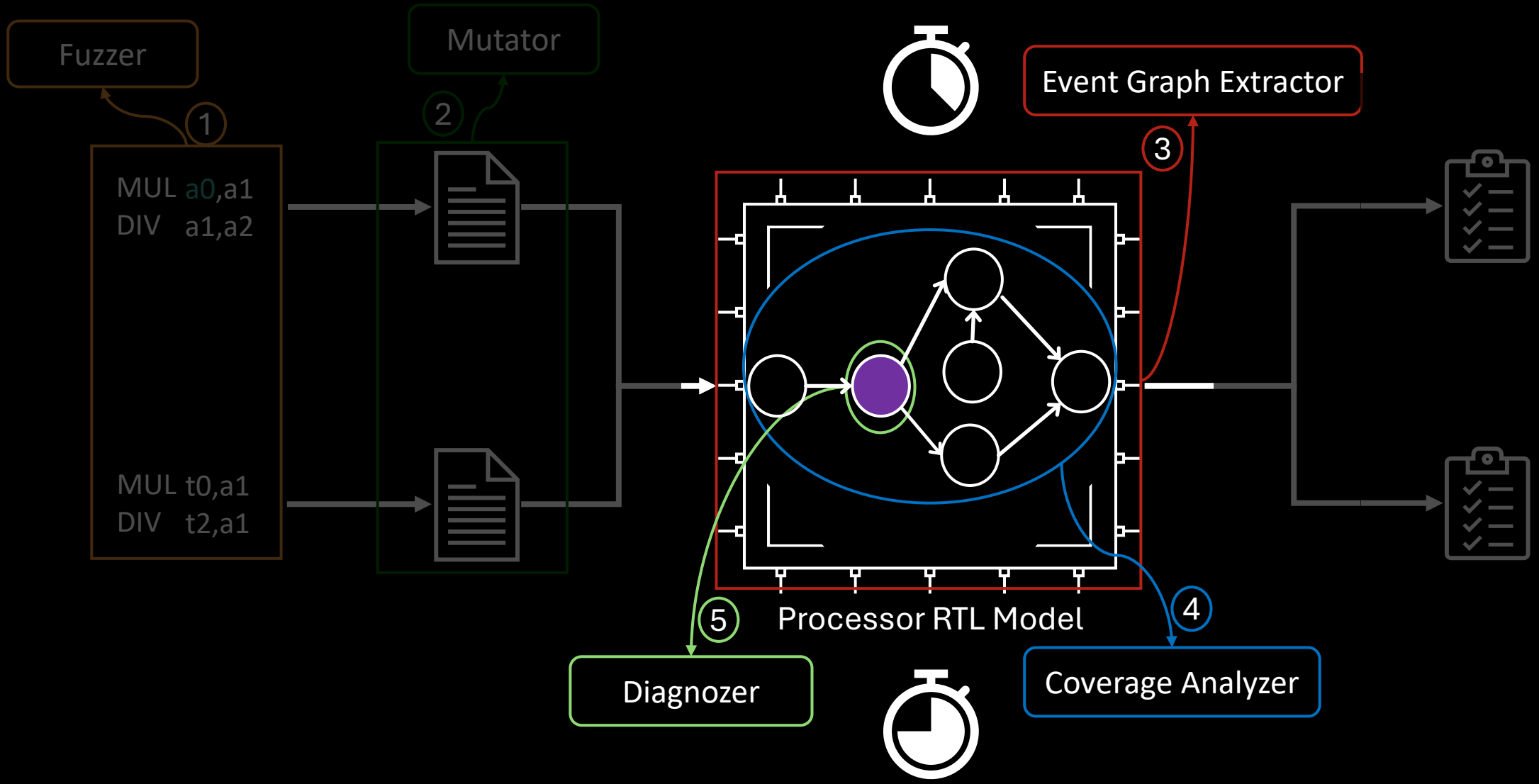
WhisperFuzz: High-Level Idea



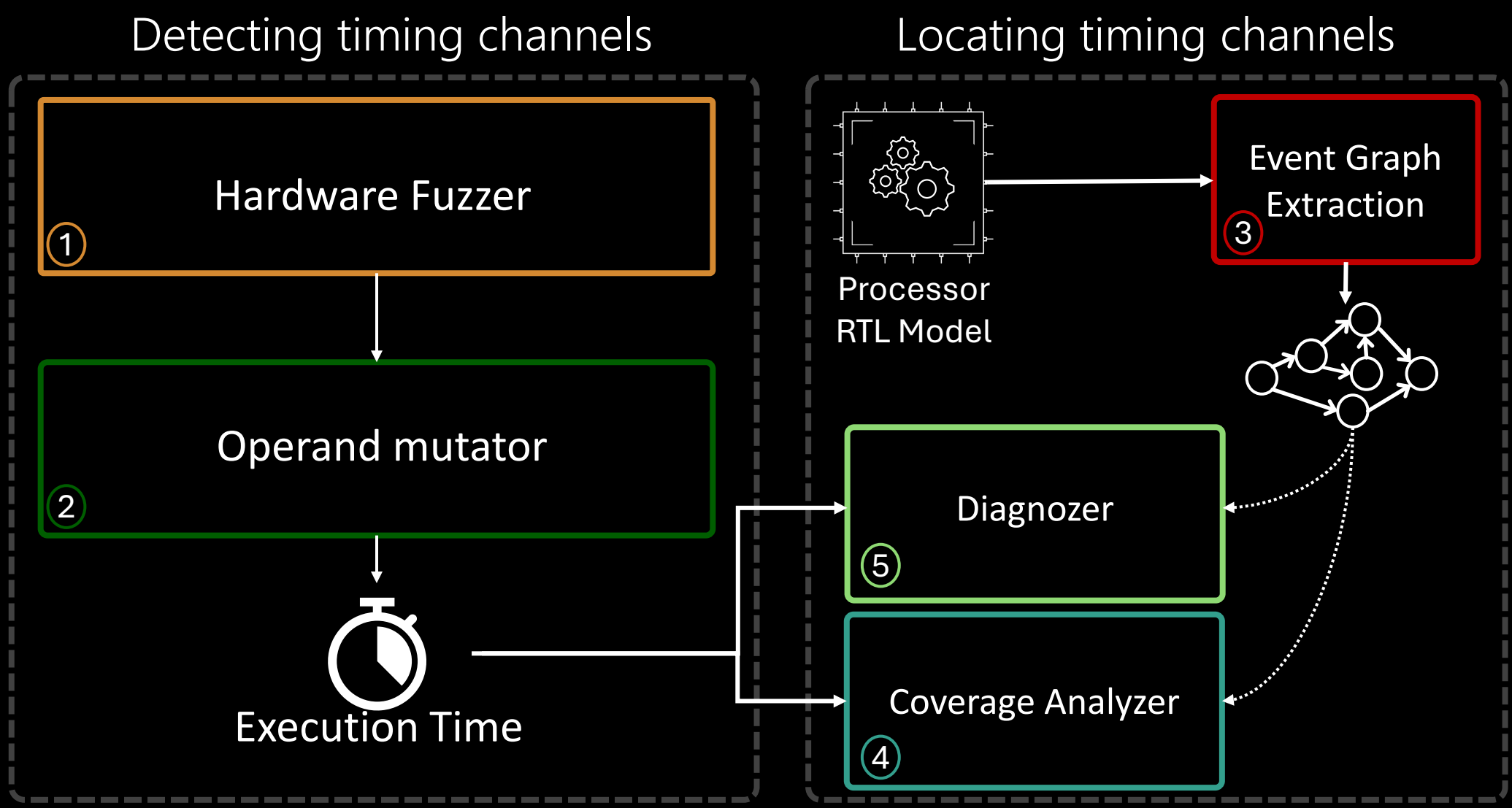
WhisperFuzz: High-Level Idea



WhisperFuzz: High-Level Idea



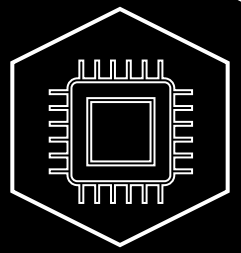
WhisperFuzz Framework



WhisperFuzz: Evaluation

Evaluation: Tested Processors

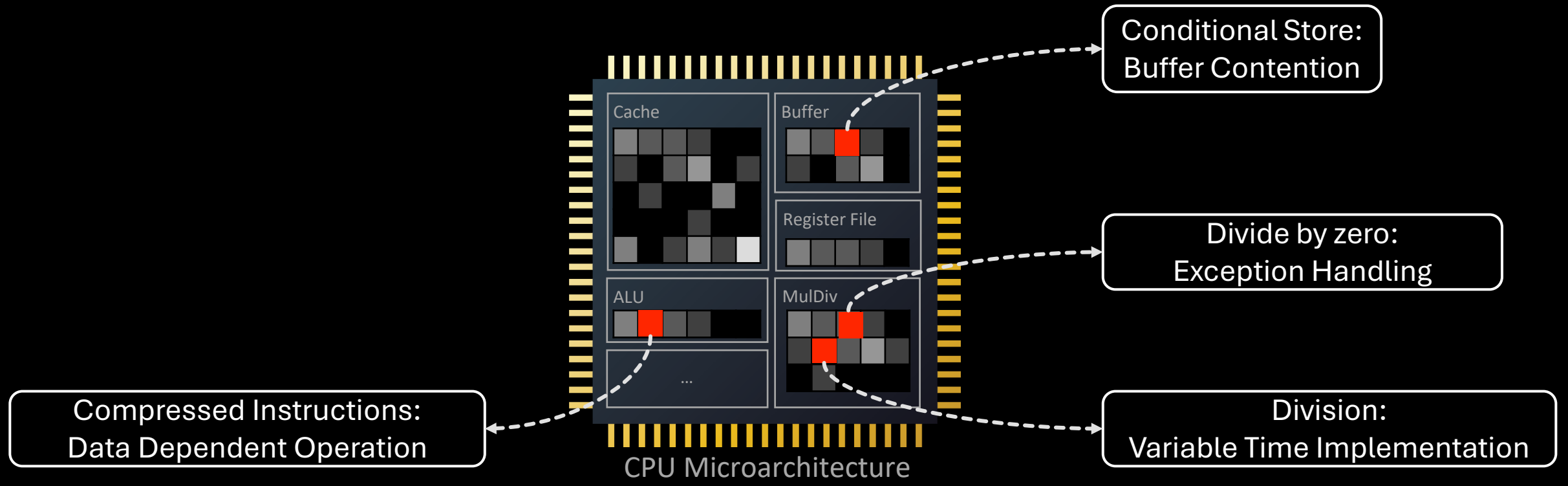
Boom: 8 stage, OutofOrder, Chisel



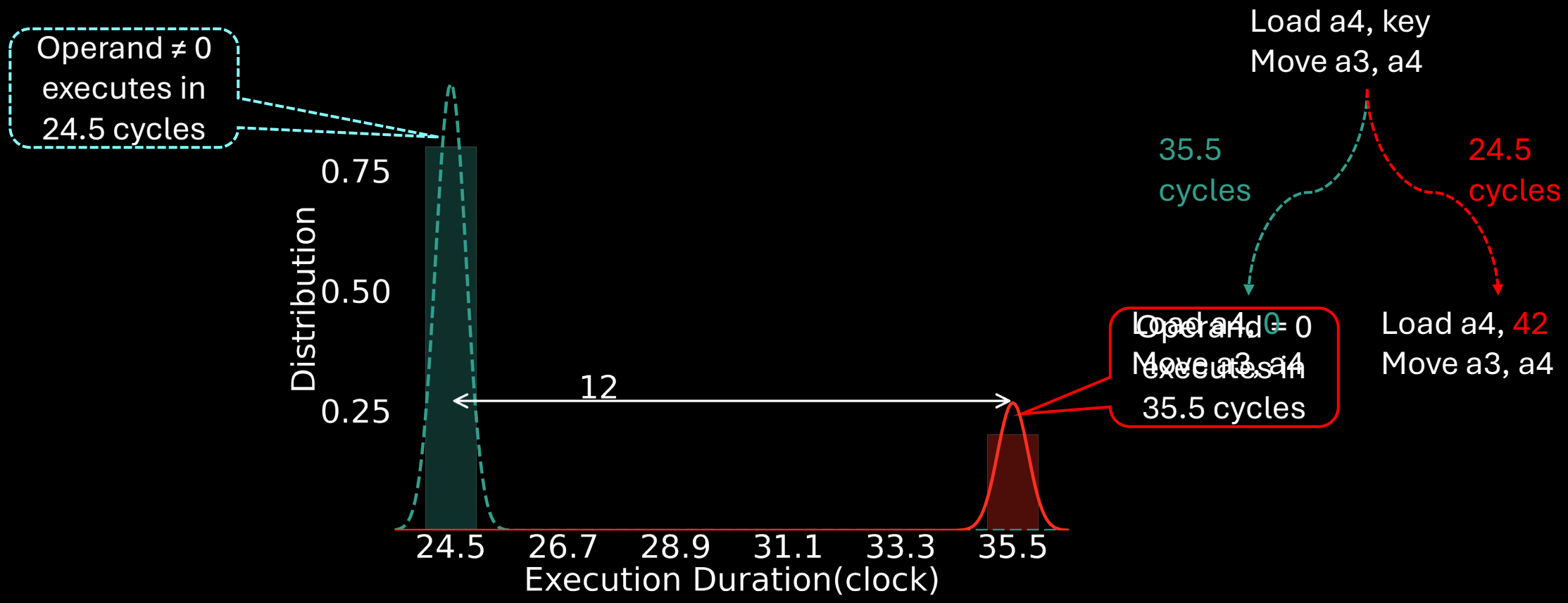
CVA6: 6 stage, InOrder, System Verilog

RocketChip: 6 stage, InOrder, Chisel

Evaluation: Leakage Sources

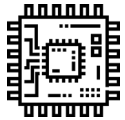


Evaluation: Novel Vulnerabilities



Execution time behavior for CVA6 compressed instruction vulnerabilities.

Key Takeaways



Timing Channels: A disastrous threat in processors

Inefficient to fix post-production

WhisperFuzz: Detects and locate these channels in designs

Novel coverage statictics

Icon Credits: Flaticon and Icons8

RSAC™ 2025 Conference

Many Voices.
One Community.

Our paper:
https://www.usenix.org/conference/usenixsecurity24/presentation/borkar